

UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE INGENIERÍA
Departamento de Informática



Modelación y Simulación
Laboratorio 1

Felipe González M.

Franco Labra L.

Profesor: Gonzalo Acuña

Ayudante: Diego Mellis

Santiago – Chile

2020

TABLA DE CONTENIDO

Índice de tablas	v
Índice de ilustraciones	vii
1 Introducción	1
2 Márco teórico	3
3 Desarrollo Primera Parte	5
4 Desarrollo Segunda Parte	9
5 Manual de uso	11
6 Conclusiones	15
Bibliografía	16

ÍNDICE DE TABLAS

ÍNDICE DE ILUSTRACIONES

Figura 3.1	Gráfico de la función a.	6
Figura 3.2	Gráfico de la función b.	7
Figura 3.3	Gráfico de las funciones a y b en conjunto.	7
Figura 3.4	Gráfico de la función c normal.	8
Figura 3.5	Gráfico de la función c logarítmica.	8
Figura 4.1	Código del algoritmo Newton Raphson.	10
Figura 5.1	Botón run en matlab.	11
Figura 5.2	Ventana de comandos.	11
Figura 5.3	Ejemplo de un correcto funcionamiento.	12
Figura 5.4	Ejemplo de cantidad de valores menor a 4.	12
Figura 5.5	Ejemplo de entradas solicitadas.	13
Figura 5.6	Solución por pantalla del ejemplo anterior.	13

CAPÍTULO 1. INTRODUCCIÓN

Este laboratorio tiene como objetivo principal, lograr acercarse mas al manejo del lenguaje interpretado de MATLAB, para esto se plantean una serie de ejercicios a realizar, que van desde la implementación de diversos gráficos de funciones comunes hasta la implementación de un algoritmo bastante conocido para aproximar la raíz de un polinomio, el algoritmo de Newton Raphson.

MATLAB es un sistema de cómputo numérico desarrollado por MathWorks. Ofrece un lenguaje de programación (lenguaje M) multiparadigmático y un entorno de desarrollo integrado. Sirve para para analizar y diseñar sistemas y productos de la realidad. Y por esto es que sus usos abarcan aprendizaje automático, procesamiento de señales, procesamiento de imágenes, visión artificial, comunicaciones, finanzas computacionales, diseño de control, robótica y muchos otros campos.”(MathWorks, 2020).

Este informe está conformado por: esta introducción; un marco teórico donde se explican los conceptos básicos de MATLAB y las funciones a representar; el desarrollo de la experiencia que se divide en gráficos de funciones e implementación de algoritmos; manual de uso; conclusiones; y referencias.

CAPÍTULO 2. MÁRCO TEÓRICO

- **MATLAB:** Corresponde a un sistema de computo numérico, con un entorno de desarrollo integrado (IDE) y con un lenguaje de programación interpretado. Combina un entorno de escritorio perfeccionado para el análisis iterativo y los procesos de diseño con un lenguaje de programación que expresa las matemáticas de matrices y arrays directamente.
- **Vector:** Los vectores en MATLAB corresponden básicamente a una representación de un conjunto de datos numéricos, también se pueden entender como arreglos.
- **Recursión:** Es la forma en la cual se especifica un proceso basado en su propia definición. Llevado a la programación es cuando una función o procedimiento está definido en términos de sí mismo, esto quiere decir que en el cuerpo de la función o procedimiento se hace un llamado a la misma función que se está definiendo.
- **Algoritmo de Newton Raphson:** Es un algoritmo para encontrar aproximaciones de los ceros o raíces de una función real. También puede ser usado para encontrar el máximo o mínimo de una función, encontrando los ceros de su primera derivada.

CAPÍTULO 3. DESARROLLO PRIMERA PARTE

Para esta experiencia se solicitaron 5 gráficos, a continuación se muestra y explica la implementación cada uno de ellos.

Para poder hacer los distintos gráficos, se comienza escribiendo `figure` antes de ellos, así, cada uno aparece en una ventana distinta. Seguido de esto, se debe usar la función `plot()` para realizar el gráfico. Esta función requiere como parámetro de entrada los valores de la abscisa y ordenada, y opcionalmente se define al final entre comillas simples que carácter se quiere usar para el gráfico y el color deseado.

- Graficar en conjunto y separado las funciones (3 gráficos):

$$a(x) = 8 \log_5(4x + 12) \quad (3.1)$$

$$b(x) = \text{sen}(6(\log_2(x + 9))) + \text{cos}(7(\log_6(4x + 32))) \quad (3.2)$$

La primera debe ser graficada en rojo y con *, mientras que la segunda debe ser graficada en verde y con +. Para poder mostrar los gráficos en conjunto, se debe definir el primero normalmente y al momento de definir el segundo se debe agregar el comando `hold on` antes de usar de nuevo la función `plot()`, y se debe usar el comando `hold off` después de esta.

- Graficar en escala normal y logarítmica la siguiente función (2 gráficos):

$$c(x) = 6e^{x+18} \quad (3.3)$$

Para este gráfico, la elección del color y estilo son a elección, pero se debe cuadrricular la figura, para esto se debe agregar `grid on` al definir el gráfico. Para poder graficar en escala logarítmica, se debe usar la función `semilogy()` en vez de `plot()`.

A continuación, se muestra cada uno de los gráficos realizados.

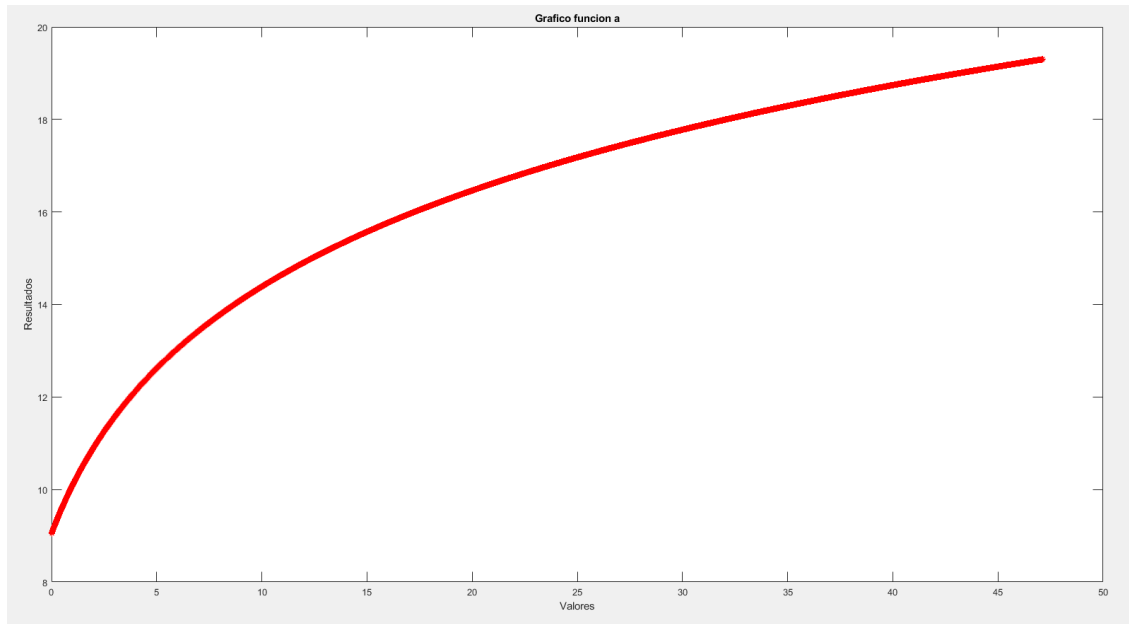


Figura 3.1: Gráfico de la función a .

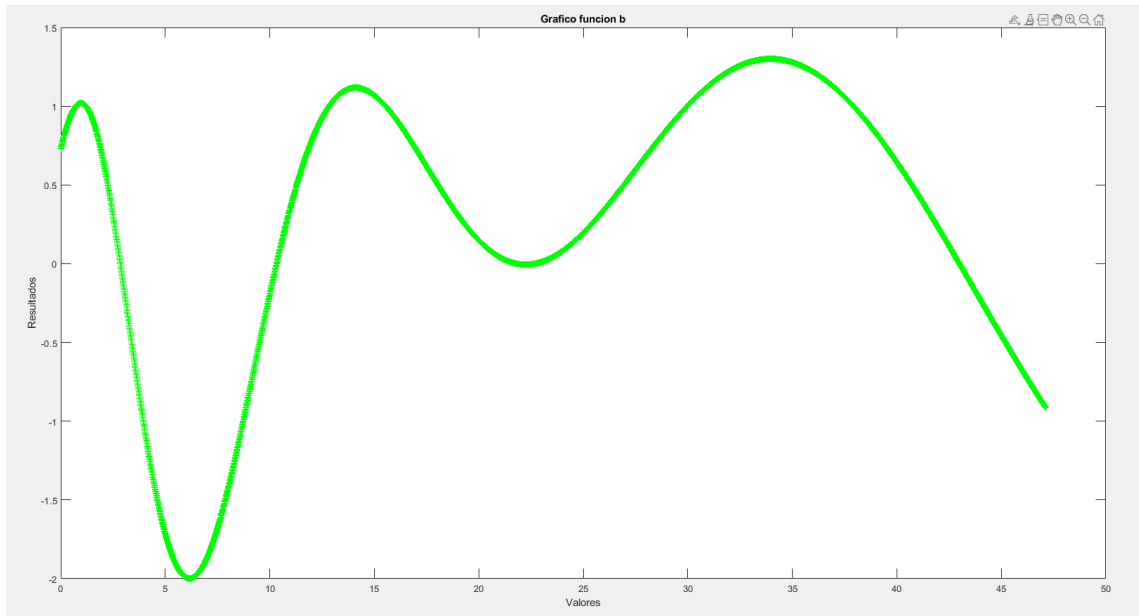


Figura 3.2: Gráfico de la función b.

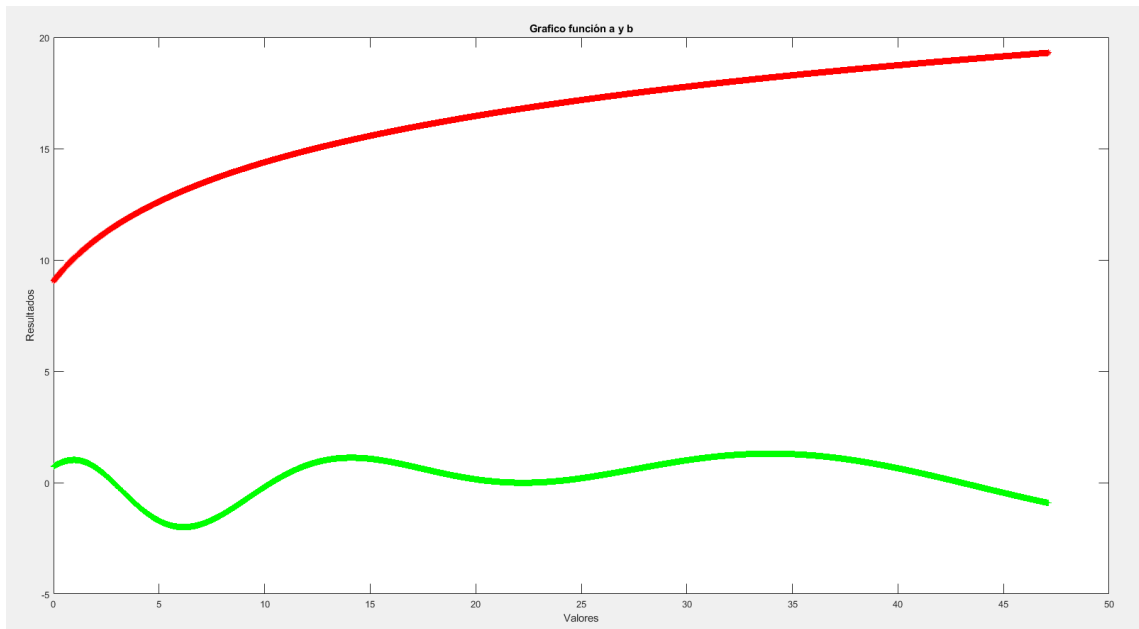


Figura 3.3: Gráfico de las funciones a y b en conjunto.

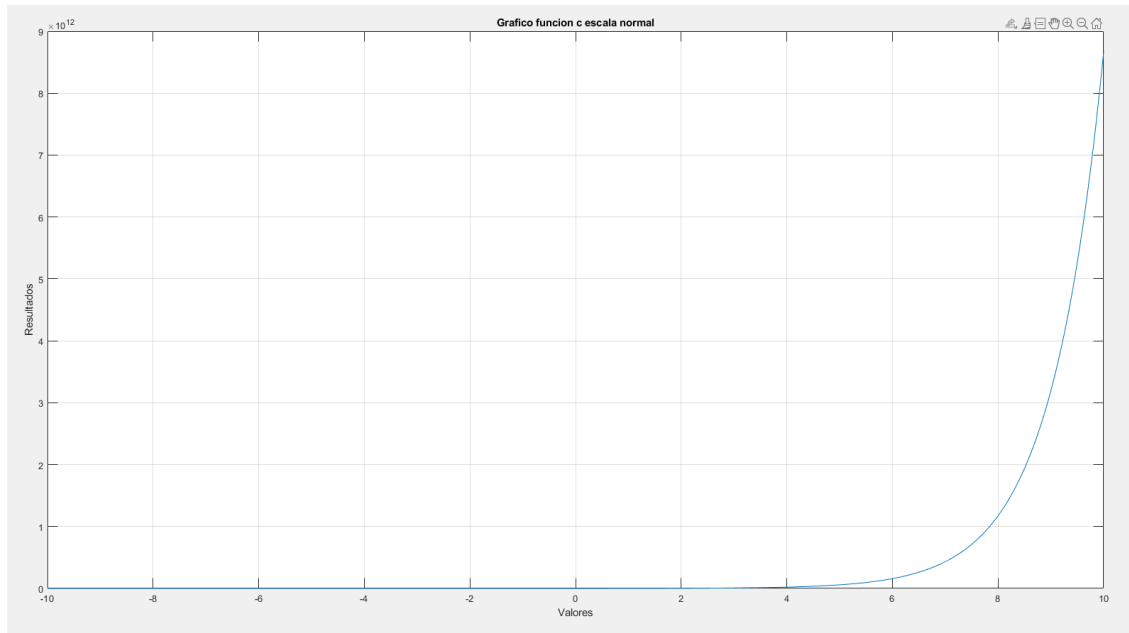


Figura 3.4: Gráfico de la función c normal.

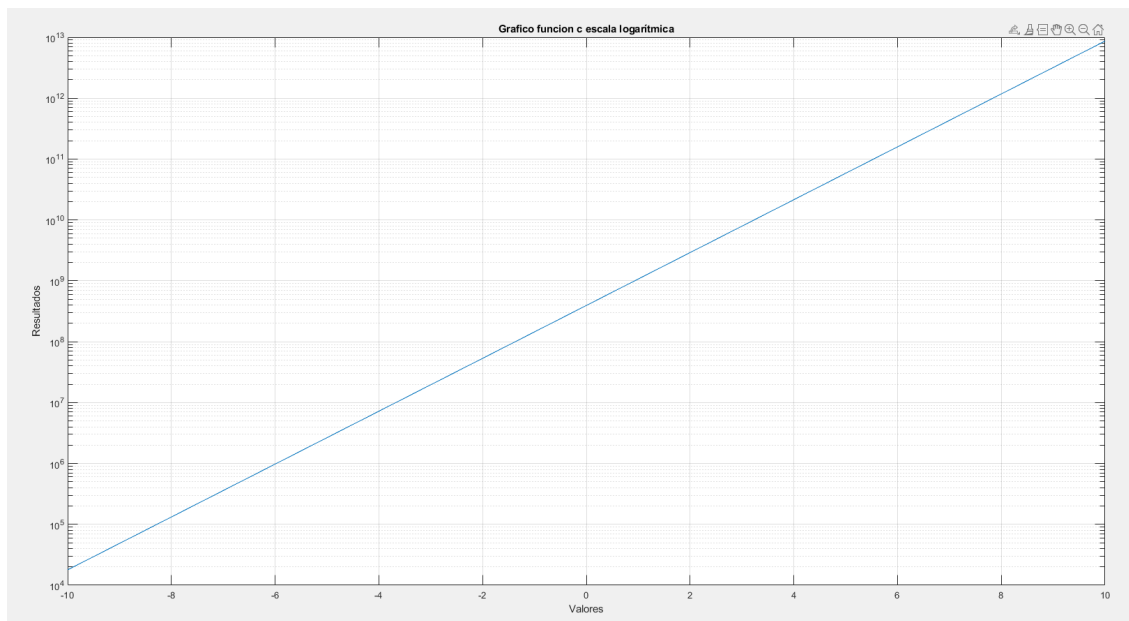


Figura 3.5: Gráfico de la función c logarítmica.

CAPÍTULO 4. DESARROLLO SEGUNDA PARTE

En este capítulo se explican los algoritmos que se implementaron para realizar esta experiencia. En la primera parte de este laboratorio, se pide graficar 3 funciones distintas. Los algoritmos que se implementaron para poder obtener los valores necesarios para graficar son bastante simples y parecidos entre sí. Para la implementación de la función a y b, se debe hacer uso de logaritmos con bases distintas a la que tiene la función nativa de Matlab, es por esto que se aplica la propiedad de cambio de base de logaritmos.

$$\log_a b = \frac{\log_c b}{\log_c a} \quad (4.1)$$

El resto de la implementación de estas funciones, corresponde básicamente a realizar las operaciones pertinentes y entregar el resultado.

Para el caso de la implementación del algoritmo Newton Raphson de forma recursiva, primero se define una función llamada "newton", que recibe los 4 parámetros de entrada que fueron solicitados en el enunciado del laboratorio (el polinomio, el numero máximo de iteraciones, el error a considerar y el valor inicial) y se encarga de obtener la derivada del polinomio de entrada, para después hacer el llamado a la función recursiva que realizara el resto del procedimiento del algoritmo. Esta función recursiva se definió como raphson", esta recibe como parámetros de entrada la función original, la derivada de la función, el numero máximo de iteraciones, el error a considerar, el valor inicial y la cantidad de iteraciones que se han realizado. El primer paso que realiza esta función, es verificar si aun no se llega al numero máximo de iteraciones, comparando el numero de iteración actual con el numero máximo que se ingresaron por parámetros de entrada, si no se ha llegado al numero máximo, se procede realizar una nueva aproximación de la raíz y posteriormente calcula el nuevo error para esta iteración. Una vez que se tiene el nuevo error, se procede a verificar si este es menor al error a considerar que se entrego por parámetro, si se cumple con esto la función termina y entrega la aproximación calculada como solución, para el caso contrario hace el llamado recursivo, entregando como valor inicial la aproximación

realizada y el numero de iteración aumentado en uno.

A continuación se muestra el algoritmo implementado.

```
function[solucion] = newton(funcion, max, error, x0)

    f = funcion;
    fl = diff(f); %derivada de la funcion
    solucion = raphson(f, fl, max, error, x0, 1); %llamado a funcion recursiva
end

function[resultado] = raphson(f, fl, max, error, x0, iteracion)

    if iteracion < max %condicion de corte de maxima iteracion
        solucion = x0 - polyval(f, x0)/polyval(fl, x0); %calculo del valor
        err = (solucion - x0) / solucion; %calculo del error

        if err > error %condicion de corte de error
            resultado = raphson(f, fl, max, error, solucion, (iteracion + 1)); %llamado recursivo
        else
            resultado = solucion;
        end
    else
        resultado = x0;
    end
end
```

Figura 4.1: Código del algoritmo Newton Raphson.

CAPÍTULO 5. MANUAL DE USO

Para poder usar el programa de forma correcta, necesita lo siguiente:

- Tener instalado en su computador el programa Matlab.
- Tener todos los archivos que vienen en el código fuente en la misma carpeta.
- Debe ejecutar el código desde el archivo con nombre "main.m".

Una vez que se tiene abierto el programa, en la parte superior se encuentra un botón de color verde que dice run", al presionar ese botón se ejecutara el programa en el archivo que se tenga abierto, como se menciona anteriormente de estar abierto el archivo "main.m".

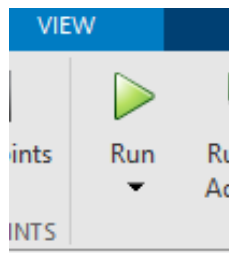


Figura 5.1: Botón run en matlab.

Cuando se presiona el botón run", los resultados y los valores que se deben ingresar saldrán en la ventana de comandos que se encuentra en la parte inferior.

```
63 - while len < 4
64 -     matriz = input('Ingrese una matriz de minimo 4 v:
65 -     len = length(matriz);
66 -     if len < 4
67 -         disp('Debe ingresar una matriz de minimo 4 v:
68 -     end

Command Window

>> main
```

Figura 5.2: Ventana de comandos.

Para el caso de la operación que se realiza al vector, se deben ingresar mínimo 4 valores numéricos, al no cumplir con esto, se le volverá a pedir los valores a calcular y se

mostrara un mensaje de que se deben ingresar mínimo 4 valores.

A continuación, se muestran los mensajes que se recibirán por la ventana de comandos en diferentes casos:

```
Ingrese una matriz de minimo 4 valores: [1]
Debe ingresar una matriz de minimo 4 valores!
Ingrese una matriz de minimo 4 valores: [1 2 3 4]

resultado =

    -2.9840
1
```

Figura 5.3: Ejemplo de un correcto funcionamiento.

```
Ingrese una matriz de minimo 4 valores: [2 3 4 5]

resultado =

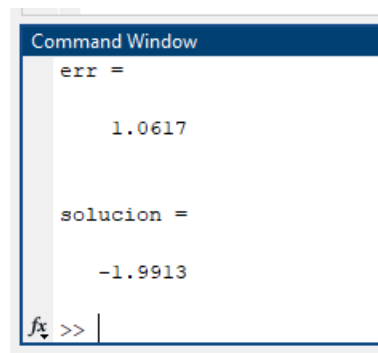
    -3.6407
```

Figura 5.4: Ejemplo de cantidad de valores menor a 4.

Cuando se desee hacer uso del algoritmo de Newton Raphson, se le van a solicitar cuatro parámetros por la consola, el polinomio del cual desea aproximar su raíz, el numero máximo de iteraciones que hará el algoritmo, el error a considerar en el calculo y el valor inicial. Una vez que se entreguen estos parámetros, se procede a realizar el calculo y se entregará por la consola la aproximación final. Debe tener en cuenta que se muestran los valores de todas las iteraciones realizadas, pero la que sale al final es la que corresponde a la que se desea. A continuación, se muestra un ejemplo del uso de esta función.

```
>> main  
Ingrese la funcion: [2 1 3]  
Ingrese el numero maximo de itearaciones: 100  
Ingrese el error a considerar: 0.0001  
Ingrese el valor inicial: 1
```

Figura 5.5: Ejemplo de entradas solicitadas.



The screenshot shows a 'Command Window' with the following output:

```
err =  
  
    1.0617  
  
solucion =  
  
   -1.9913  
fx >> |
```

Figura 5.6: Solución por pantalla del ejemplo anterior.

CAPÍTULO 6. CONCLUSIONES

A modo de conclusión, es posible decir que se cumplió el objetivo principal en su totalidad. Esto queda en evidencia en lo plasmado en este informe: se logró retratar los procesos realizados del desarrollo de los ejercicios propuestos en esta experiencia. Lo anterior también nos permitió poner a prueba las capacidades y el aprendizaje obtenido en base a la información entregada en el enunciado, así como también las investigaciones propias realizadas para lograr implementar lo solicitado.

Al momento de llevar a cabo la experiencia, no se presentaron problemas ni bloqueos que entorpecieran el proceso de aprendizaje, si se puede destacar que la implementación del algoritmo de Newton Raphson llevo un poco mas de tiempo para implementar, ya que existía la incertidumbre de si el resultado entregado era correcto, pero posteriormente se logro realizar el código del algoritmo pertinente de manera correcta.

Finalmente, se agrega que la experiencia se desarrolló de manera satisfactoria, por lo que se espera el mismo resultado en las próximas experiencias de la asignatura.

BIBLIOGRAFÍA

wikipedia (2020a). Matlab. Recuperado desde <https://es.wikipedia.org/wiki/MATLAB>".

wikipedia (2020b). Método de newton raphson. "<https://es.wikipedia.org/wiki/M>

wikipedia (2020c). Recursión. Recuperado desde <https://es.wikipedia.org/wiki/Recursi%C3%B3n>".